# Part I - (Loan Data from Prosper)

## by (Afnan Abdullah K Alshehri)

## Introduction

> This data set contains 113,937 loans with 81 variables on each loan, including loan amount, borrower rate (or interest rate), current loan status, borrower income, and many others.

## Preliminary Wrangling

In [184…

```
# import all packages and set plots to be embedded inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

%matplotlib inline
```

In [185…

```
#Load dataset
loan=pd.read_csv(r"C:\Users\HP\.jupyter\prosperLoanData.csv")
pd.set_option("display.max_rows", None,"display.max_columns", None) #show entire datafra
loan.head()
```

Out[185]:

| | ListingKey | ListingNumber | ListingCreationDate | CreditGrade | Term | LoanStatus | ClosedDate | Bo |
|---|---|---|---|---|---|---|---|---|
| **0** | 1021339766868145413AB3B | 193129 | 2007-08-26 19:09:29.263000000 | C | 36 | Completed | 2009-08-14 00:00:00 | |
| **1** | 10273602499503308B223C1 | 1209647 | 2014-02-27 08:28:07.900000000 | NaN | 36 | Current | NaN | |
| **2** | 0EE9337825851032864889A | 81716 | 2007-01-05 15:00:47.090000000 | HR | 36 | Completed | 2009-12-17 00:00:00 | |
| **3** | 0EF5356002482715299901A | 658116 | 2012-10-22 11:02:35.010000000 | NaN | 36 | Current | NaN | |
| **4** | 0F023589499656230C5E3E2 | 909464 | 2013-09-14 18:38:39.097000000 | NaN | 36 | Current | NaN | |

In [186…

```
loan.isnull().sum()
```

Out[186]:

```
ListingKey                          0
ListingNumber                       0
ListingCreationDate                 0
CreditGrade                     84984
Term                                0
LoanStatus                          0
ClosedDate                      58848
BorrowerAPR                        25
BorrowerRate                        0
LenderYield                         0
EstimatedEffectiveYield         29084
EstimatedLoss                   29084
```

```
EstimatedReturn                         29084
ProsperRating (numeric)                 29084
ProsperRating (Alpha)                   29084
ProsperScore                            29084
ListingCategory (numeric)                   0
BorrowerState                            5515
Occupation                               3588
EmploymentStatus                         2255
EmploymentStatusDuration                 7625
IsBorrowerHomeowner                         0
CurrentlyInGroup                            0
GroupKey                               100596
DateCreditPulled                            0
CreditScoreRangeLower                     591
CreditScoreRangeUpper                     591
FirstRecordedCreditLine                   697
CurrentCreditLines                       7604
OpenCreditLines                          7604
TotalCreditLinespast7years                697
OpenRevolvingAccounts                       0
OpenRevolvingMonthlyPayment                 0
InquiriesLast6Months                      697
TotalInquiries                           1159
CurrentDelinquencies                      697
AmountDelinquent                         7622
DelinquenciesLast7Years                   990
PublicRecordsLast10Years                  697
PublicRecordsLast12Months                7604
RevolvingCreditBalance                   7604
BankcardUtilization                      7604
AvailableBankcardCredit                  7544
TotalTrades                              7544
TradesNeverDelinquent (percentage)       7544
TradesOpenedLast6Months                  7544
DebtToIncomeRatio                        8554
IncomeRange                                 0
IncomeVerifiable                            0
StatedMonthlyIncome                         0
LoanKey                                     0
TotalProsperLoans                       91852
TotalProsperPaymentsBilled              91852
OnTimeProsperPayments                   91852
ProsperPaymentsLessThanOneMonthLate     91852
ProsperPaymentsOneMonthPlusLate         91852
ProsperPrincipalBorrowed                91852
ProsperPrincipalOutstanding             91852
ScorexChangeAtTimeOfListing             95009
LoanCurrentDaysDelinquent                   0
LoanFirstDefaultedCycleNumber           96985
LoanMonthsSinceOrigination                  0
LoanNumber                                  0
LoanOriginalAmount                          0
LoanOriginationDate                         0
LoanOriginationQuarter                      0
MemberKey                                   0
MonthlyLoanPayment                          0
LP_CustomerPayments                         0
LP_CustomerPrincipalPayments                0
LP_InterestandFees                          0
LP_ServiceFees                              0
LP_CollectionFees                           0
LP_GrossPrincipalLoss                       0
LP_NetPrincipalLoss                         0
LP_NonPrincipalRecoverypayments             0
PercentFunded                               0
Recommendations                             0
```

```
InvestmentFromFriendsCount                  0
InvestmentFromFriendsAmount                 0
Investors                                   0
dtype: int64
```

In [187…

```
loan_n=loan[['ListingKey','ListingCreationDate','Term','LoanStatus','BorrowerAPR','Prosp
             'ListingCategory (numeric)','BorrowerState','EmploymentStatus','IsBorrowerH
             'OpenCreditLines','IncomeRange','StatedMonthlyIncome','TotalProsperLoans',
             'LoanMonthsSinceOrigination','LoanOriginalAmount','LoanOriginationQuarter',
             'MonthlyLoanPayment','Investors']].copy()
```

In [188…

```
loan_n.head()
```

Out[188]:

| | ListingKey | ListingCreationDate | Term | LoanStatus | BorrowerAPR | ProsperRating (Alpha) | ListingCategor (numeric |
|---|---|---|---|---|---|---|---|
| **0** | 1021339766868145413AB3B | 2007-08-26 19:09:29.263000000 | 36 | Completed | 0.16516 | NaN | |
| **1** | 10273602499503308B223C1 | 2014-02-27 08:28:07.900000000 | 36 | Current | 0.12016 | A | |
| **2** | 0EE9337825851032864889A | 2007-01-05 15:00:47.090000000 | 36 | Completed | 0.28269 | NaN | |
| **3** | 0EF5356002482715299901A | 2012-10-22 11:02:35.010000000 | 36 | Current | 0.12528 | A | 1 |
| **4** | 0F023589499656230C5E3E2 | 2013-09-14 18:38:39.097000000 | 36 | Current | 0.24614 | D | |

In [189…

```
loan_n.shape
```

Out[189]:

```
(113937, 19)
```

In [190…

```
loan_n.sample(10)
```

Out[190]:

| | ListingKey | ListingCreationDate | Term | LoanStatus | BorrowerAPR | ProsperRating (Alpha) | ListingC (n |
|---|---|---|---|---|---|---|---|
| **75765** | 7C033601306041319C7B6D6 | 2014-01-31 14:42:13.940000000 | 36 | Current | 0.09434 | AA | |
| **85583** | 4F573587665343589DD5917 | 2013-09-07 09:52:31.880000000 | 36 | Past Due (61-90 days) | 0.17601 | B | |
| **80185** | CEEA3408263030808942908 | 2007-12-15 10:50:35.270000000 | 36 | Completed | 0.07469 | NaN | |
| **108227** | E3B23378355908760696787 | 2006-12-26 14:42:20.433000000 | 36 | Completed | 0.22248 | NaN | |
| **20559** | BCD935166462043806A6F22 | 2011-06-04 19:10:26.457000000 | 36 | Chargedoff | 0.22362 | C | |
| **110205** | 82B23472202052757300F47 | 2010-01-05 18:57:57.053000000 | 36 | Completed | 0.10436 | A | |
| **40766** | 7AEF3570852372317141E3B | 2013-02-20 10:31:41.383000000 | 60 | Current | 0.33040 | E | |
| **93784** | BBFE3513449501325FF1B56 | 2011-04-15 08:30:48.210000000 | 36 | Completed | 0.10375 | A | |
| **78328** | 872533662393075652 51A9C | 2006-06-24 13:43:39.680000000 | 36 | Defaulted | 0.19730 | NaN | |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| **7578** | 5B9533647189373072AFCA5 | 2006-07-08 09:53:16.153000000 | 36 | Completed | 0.08684 | NaN |

```
In [191…  loan_n.describe()
```

Out[191]:

|  | Term | BorrowerAPR | ListingCategory (numeric) | OpenCreditLines | StatedMonthlyIncome | TotalProsperLoans |
|---|---|---|---|---|---|---|
| **count** | 113937.000000 | 113912.000000 | 113937.000000 | 106333.000000 | 1.139370e+05 | 22085.000000 |
| **mean** | 40.830248 | 0.218828 | 2.774209 | 9.260164 | 5.608026e+03 | 1.421100 |
| **std** | 10.436212 | 0.080364 | 3.996797 | 5.022644 | 7.478497e+03 | 0.764042 |
| **min** | 12.000000 | 0.006530 | 0.000000 | 0.000000 | 0.000000e+00 | 0.000000 |
| **25%** | 36.000000 | 0.156290 | 1.000000 | 6.000000 | 3.200333e+03 | 1.000000 |
| **50%** | 36.000000 | 0.209760 | 1.000000 | 9.000000 | 4.666667e+03 | 1.000000 |
| **75%** | 36.000000 | 0.283810 | 3.000000 | 12.000000 | 6.825000e+03 | 2.000000 |
| **max** | 60.000000 | 0.512290 | 20.000000 | 54.000000 | 1.750003e+06 | 8.000000 |

```
In [192…  loan_n.isnull().sum()

Out[192]: ListingKey                        0
          ListingCreationDate               0
          Term                              0
          LoanStatus                        0
          BorrowerAPR                      25
          ProsperRating (Alpha)         29084
          ListingCategory (numeric)         0
          BorrowerState                  5515
          EmploymentStatus               2255
          IsBorrowerHomeowner               0
          OpenCreditLines                7604
          IncomeRange                       0
          StatedMonthlyIncome               0
          TotalProsperLoans             91852
          LoanMonthsSinceOrigination        0
          LoanOriginalAmount                0
          LoanOriginationQuarter            0
          MonthlyLoanPayment                0
          Investors                         0
          dtype: int64

In [193…  loan_n.ListingKey.duplicated().sum()

Out[193]: 871

In [194…  loan_n.drop_duplicates(subset = 'ListingKey', inplace = True)

In [195…  loan_n.duplicated().sum()

Out[195]: 0

In [196…  loan_n.info()

          <class 'pandas.core.frame.DataFrame'>
          Int64Index: 113066 entries, 0 to 113936
          Data columns (total 19 columns):
           #   Column                          Non-Null Count   Dtype
```

```
 ---  ------                       --------------   -----
 0    ListingKey                   113066 non-null  object
 1    ListingCreationDate          113066 non-null  object
 2    Term                         113066 non-null  int64
 3    LoanStatus                   113066 non-null  object
 4    BorrowerAPR                  113041 non-null  float64
 5    ProsperRating (Alpha)        83982 non-null   object
 6    ListingCategory (numeric)    113066 non-null  int64
 7    BorrowerState                107551 non-null  object
 8    EmploymentStatus             110811 non-null  object
 9    IsBorrowerHomeowner          113066 non-null  bool
 10   OpenCreditLines              105462 non-null  float64
 11   IncomeRange                  113066 non-null  object
 12   StatedMonthlyIncome          113066 non-null  float64
 13   TotalProsperLoans            21923 non-null   float64
 14   LoanMonthsSinceOrigination   113066 non-null  int64
 15   LoanOriginalAmount           113066 non-null  int64
 16   LoanOriginationQuarter       113066 non-null  object
 17   MonthlyLoanPayment           113066 non-null  float64
 18   Investors                    113066 non-null  int64
dtypes: bool(1), float64(5), int64(5), object(8)
memory usage: 16.5+ MB
```

In [197…

```
#change object type in dates to to datatype
def dates(x):
    loan_n[x]=pd.to_datetime(loan[x])
dates('ListingCreationDate')
```

In [198…

```
#fill nulls values with 0
def nulls_i(x):
    loan[x]=loan_n[x].fillna(0, inplace=True)
nulls_i('OpenCreditLines')
nulls_i('TotalProsperLoans')
nulls_i('BorrowerAPR')
```

In [199…

```
def integers(x):
    loan_n[x]=loan_n[x].astype('int')
integers('OpenCreditLines')
integers('TotalProsperLoans')
```

In [200…

```
past_due=loan_n['LoanStatus'].str.contains("Past Due")

loan_n.loc[past_due,'LoanStatus']='Past Due'
```

In [201…

```
#loan_n['ProsperRating (Alpha)'] = loan_n['ProsperRating (Alpha)'].fillna('Not Available
loan_n['BorrowerState'] = loan_n['BorrowerState'].fillna('N/A')
loan_n.isnull().sum()
```

Out[201]:

```
ListingKey                       0
ListingCreationDate              0
Term                             0
LoanStatus                       0
BorrowerAPR                      0
ProsperRating (Alpha)        29084
ListingCategory (numeric)        0
BorrowerState                    0
EmploymentStatus              2255
IsBorrowerHomeowner              0
OpenCreditLines                  0
IncomeRange                      0
StatedMonthlyIncome              0
TotalProsperLoans                0
LoanMonthsSinceOrigination       0
LoanOriginalAmount               0
```

```
LoanOriginationQuarter          0
MonthlyLoanPayment              0
Investors                       0
dtype: int64
```

In [202…

```
#Change labels in Listing Category from numbers to clear list
labels={0:'Not Available', 1:'Debt Consolidation',2:'Home Improvement',3:'Business',4:'P
        5:'Student Use',6:'Auto',7:'Other',8:'Baby&Adoption',9:'Boat',10:'Cosmetic Proce
        11:'Engagement Ring',12:'Green Loans',13:'Household Expenses',14:'Large Purchase
        15:'Medical/Dental',16:'Motorcycle',17:'RV',18:'Taxes',19:'Vacation',20:'Wedding
def Listing_Category(x):
    if x in list(labels.keys()):
        return (labels[x])
    else:
        return ('Not found')


loan_n['ListingCategory (numeric)']=loan_n['ListingCategory (numeric)'].apply(Listing_Ca
```

In [203…

```
def category(x):
    loan_n[x]=loan_n[x].astype('category')
category('Term')
category('LoanStatus')
category('ListingCategory (numeric)')
category('ProsperRating (Alpha)')
category('BorrowerState')
category('EmploymentStatus')
category('IncomeRange')
```

In [204…

```
loan_n.rename(columns={'ListingKey':'Listing Key','ListingCreationDate':'Listing Creatio
                       'BorrowerAPR':'Borrower APR','ProsperRating (Alpha)':'Prosper Rat
                       'BorrowerState':'Borrower State','EmploymentStatus':'Employment S
                       'IsBorrowerHomeowner':'Is Borrower Homeowner','OpenCreditLines':'
                       'StatedMonthlyIncome':'Stated Monthly Income','TotalProsperLoans'
                       'LoanMonthsSinceOrigination':'Loan Months Since Origination','Loa
                       'LoanOriginationQuarter':'Loan Origination Quarter','MonthlyLoanP
```

In [205…

```
loan_n.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 113066 entries, 0 to 113936
Data columns (total 19 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   Listing Key                    113066 non-null  object
 1   Listing Creation Date          113066 non-null  datetime64[ns]
 2   Term                           113066 non-null  category
 3   Loan Status                    113066 non-null  category
 4   Borrower APR                   113066 non-null  float64
 5   Prosper Rating                 83982 non-null   category
 6   Listing Category               113066 non-null  category
 7   Borrower State                 113066 non-null  category
 8   Employment Status              110811 non-null  category
 9   Is Borrower Homeowner          113066 non-null  bool
 10  Open Credit Lines              113066 non-null  int32
 11  Income Range                   113066 non-null  category
 12  Stated Monthly Income          113066 non-null  float64
 13  Total Prosper Loans            113066 non-null  int32
 14  Loan Months Since Origination  113066 non-null  int64
 15  Loan Original Amount           113066 non-null  int64
 16  Loan Origination Quarter       113066 non-null  object
 17  Monthly Loan Payment           113066 non-null  float64
 18  Investors                      113066 non-null  int64
dtypes: bool(1), category(7), datetime64[ns](1), float64(3), int32(2), int64(3), object
```

```
(2)
memory usage: 10.4+ MB
```

In [ ]:

```

```

## What is the structure of your dataset?

> There are 113937 of rows and 19 of columns ,most of the variables are numeric, and some of them categorical variables.

## What is/are the main feature(s) of interest in your dataset?

> I would like to discover what are the major features for predicting the borrower annual percentage rate of loan in the dataset.

## What features in the dataset do you think will help support your investigation into your feature(s) of interest?

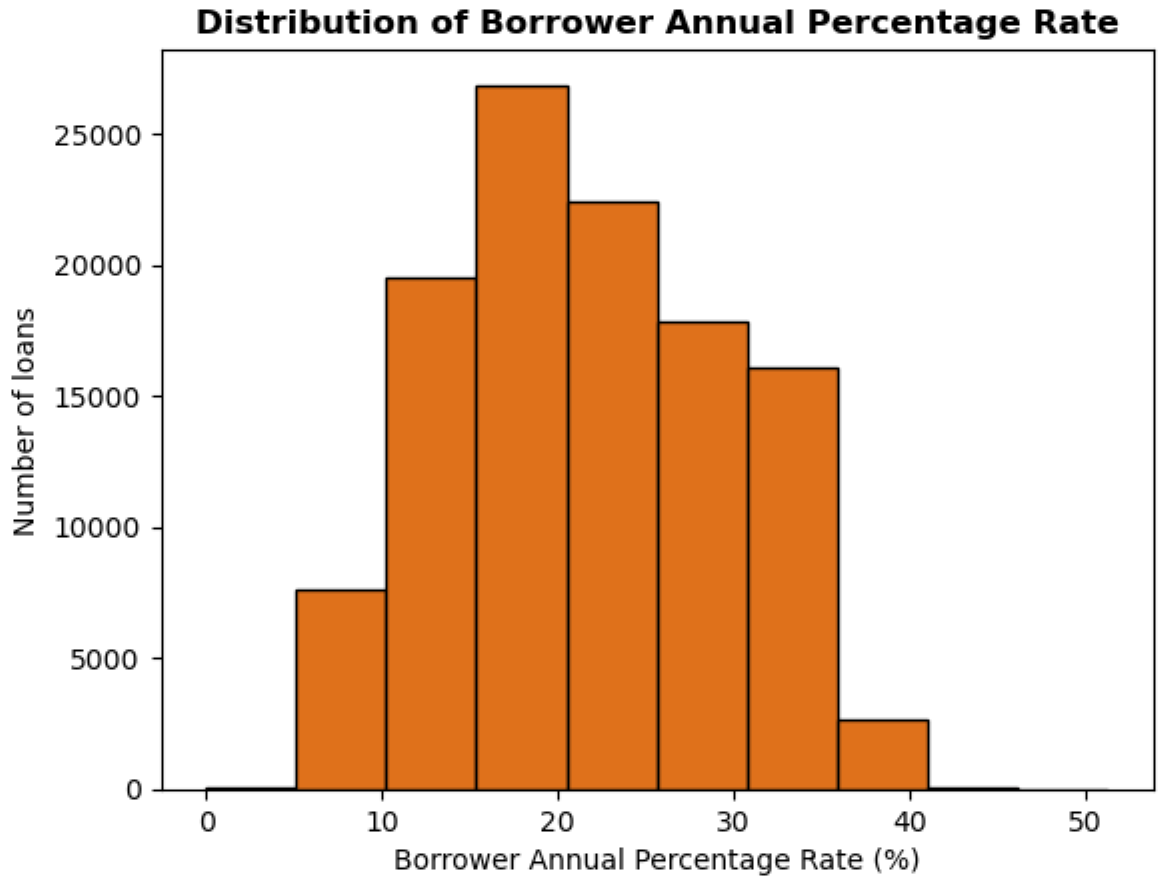
> I think the Term,Loan Status,Borrower APR,Prosper Rating,Listing Category,Borrower State,Employment Status,Income Range,Loan Original Amount,Monthly Loan Payment,Investors the most helpful features to help my go through the investigation part in our dataset

## Univariate Exploration

## Q1: What is distribution of Borrower Annual Percentage Rate?

In [206…

```
loan_n['BorrowerAPR_percent']=loan_n['Borrower APR'].mul(100)
plt.hist(data=loan_n,x='BorrowerAPR_percent',color='#DF711B',edgecolor='black')
#plt.xscale('log')
plt.xlabel('Borrower Annual Percentage Rate (%)')
plt.ylabel('Number of loans')
plt.title(f'Distribution of Borrower Annual Percentage Rate',weight='bold');
```

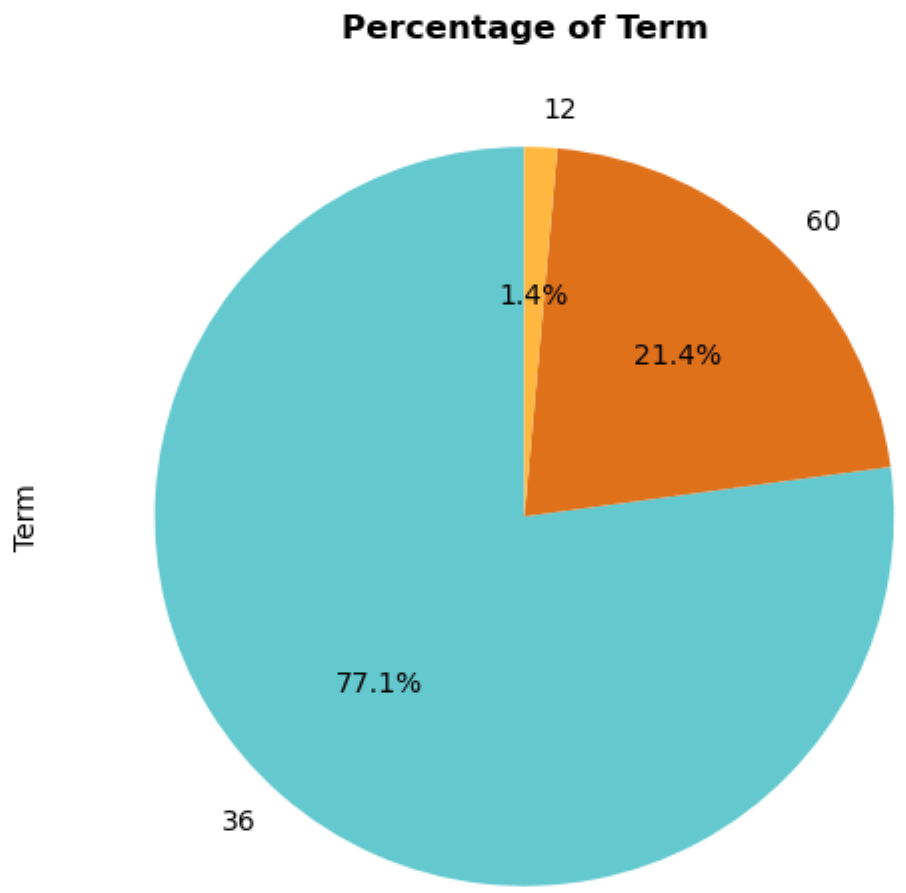


The most percentage in Borrower Annual Percentage Rate is 20 %

## Q2: What is the most length of the loan expressed in months?

In [207…

```
def piePlot(x):
    c=['#64C9CF','#DF711B','#FFB740']
    loan_n[x].value_counts(normalize=True).mul(100).plot.pie(colors=c,figsize=(8,6),
    autopct='%1.1f%%',startangle=90);
    plt.title(f'Percentage of {x}',weight='bold');


piePlot('Term');
```

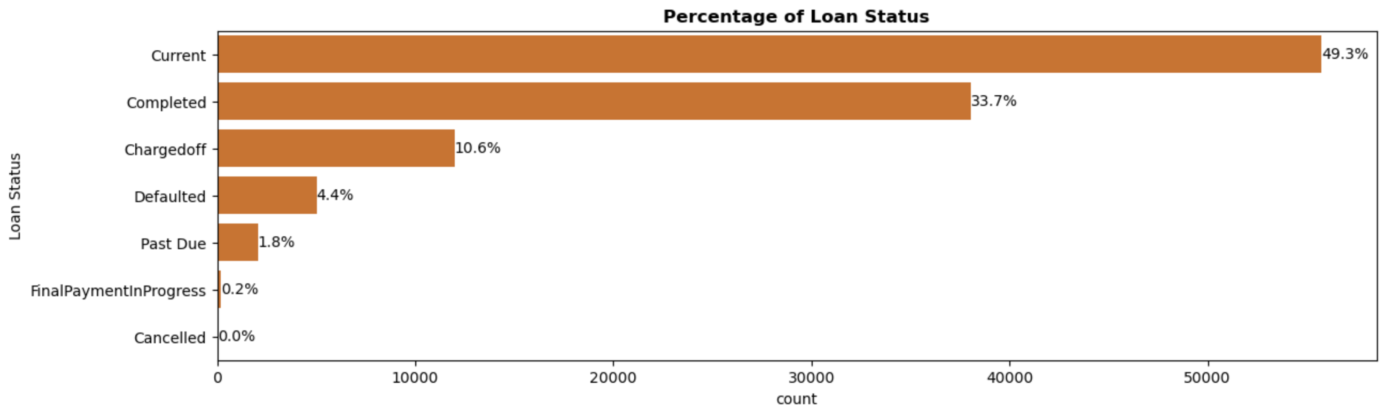


The most length of the loan is 36 months with 77%

## Q3: What is the most status of the loan?

In [208…

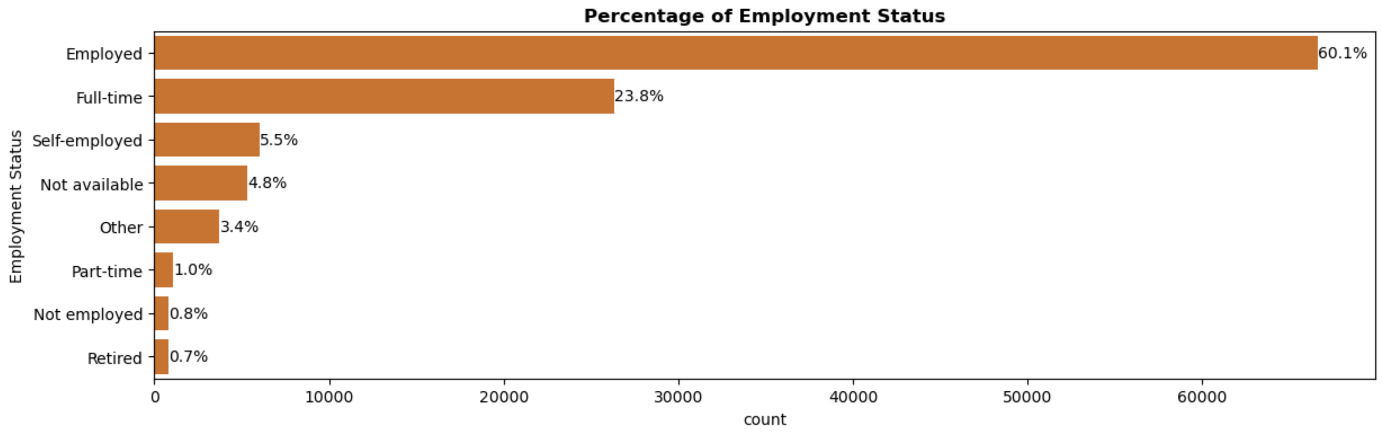
```
def bar_h(x):

    plt.figure(figsize=[14,4])
    order1=loan_n[x].value_counts().index
    sb.countplot(data=loan_n,y=x,color='#DF711B',order=order1)
    plt.title(f'Percentage of {x}',weight='bold')

    count_x=loan_n[x].value_counts()
    n_x=loan_n[x].value_counts().sum()
    for i in range(count_x.shape[0]):  ##عدد الانواع
        count=count_x[i]
        per='{:0.1f}%'.format(count*100/n_x)
        plt.text(count+1,i,per,va='center')

bar_h('Loan Status')
```



The most status of the loan is Current with 49%

## Q4: What is the most employment status of the borrower?

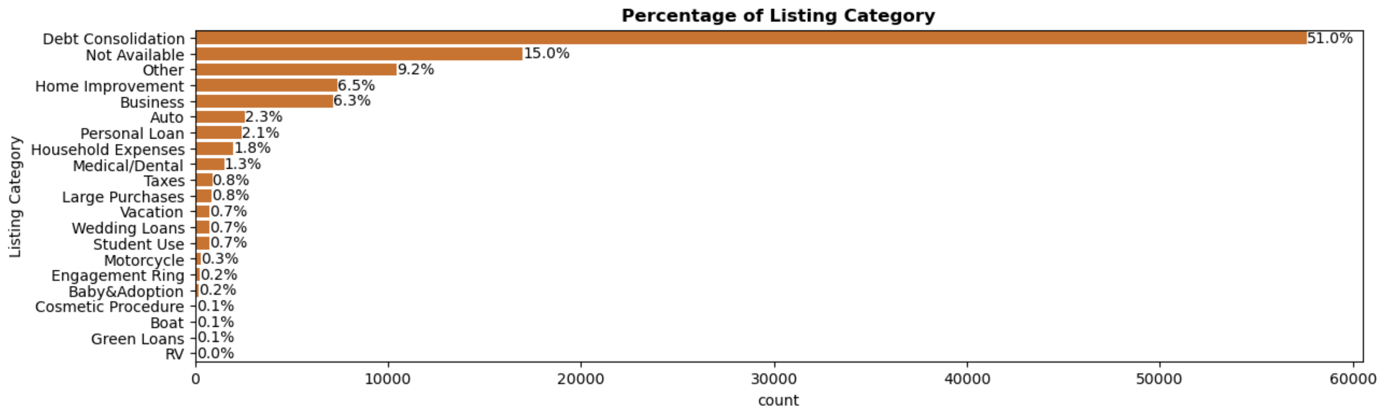
In [209…
```
bar_h('Employment Status')
```



The most employment status of the borrowers is employed with 60% then full-time with 23%

## Q5: What is the category of the listing that the borrower selected?

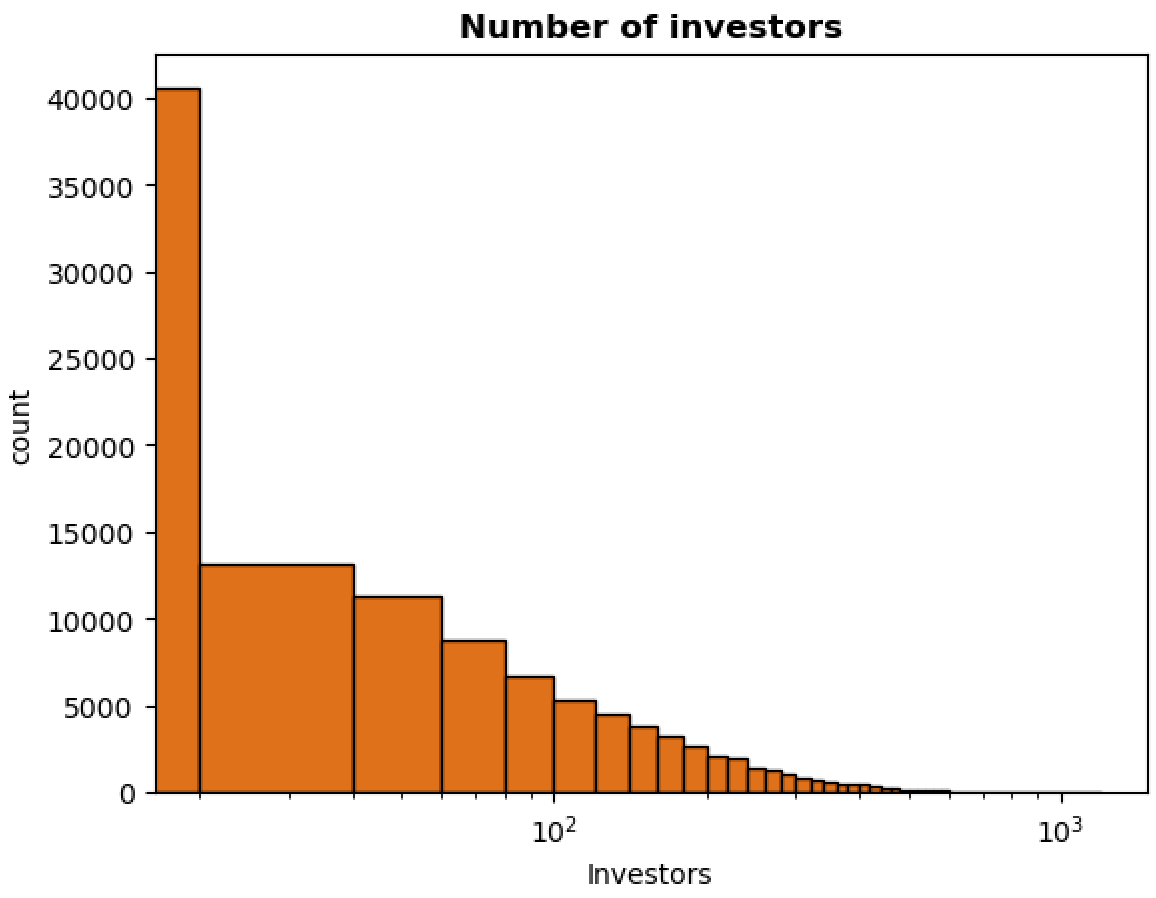
In [210…
```
bar_h('Listing Category')
```



50% of category of the listing that the borrower selected is Debt Consolidation

## Q6: What is most number of investors?

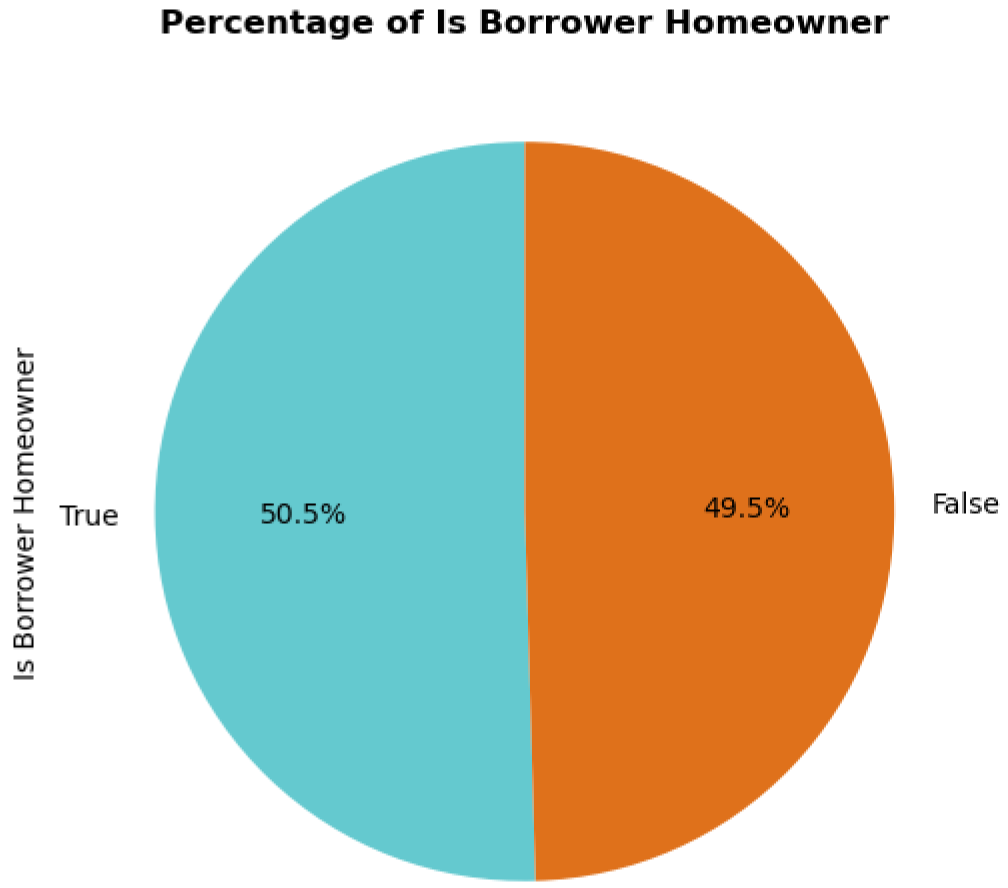
In [211…
```
#plt.figure(figsize=[14,4])
bins=np.arange(0,loan_n['Investors'].max()+20,20)
plt.hist(data=loan_n,x='Investors',bins=bins,color='#DF711B',edgecolor='black')
plt.xscale('log')
plt.xlabel('Investors')
plt.ylabel('count')
plt.title(f'Number of investors',weight='bold');
```



**Most investors under 100**

## Q7: What is percentage of borrower homeowner?

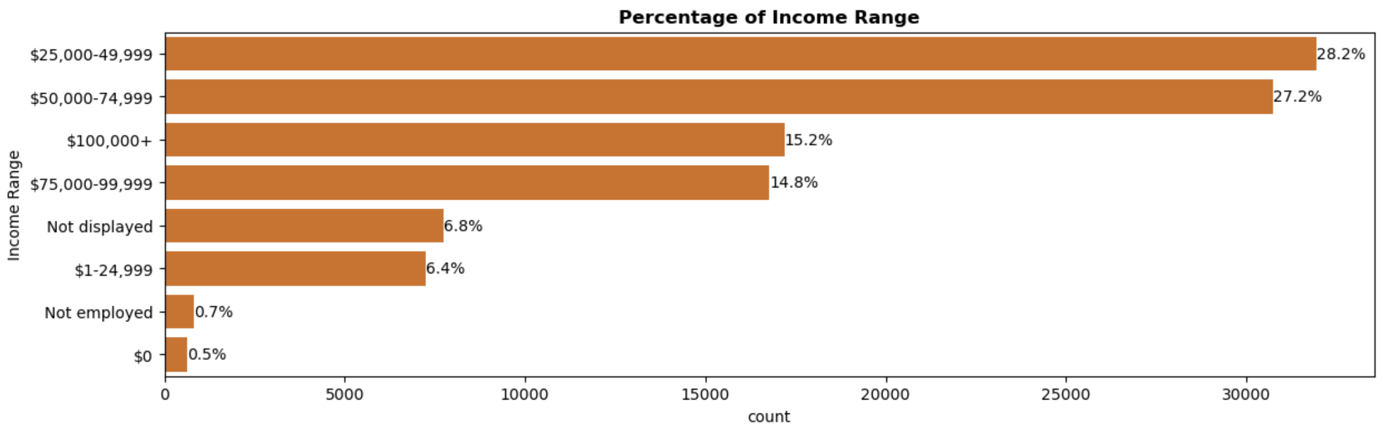
In [212…
```
piePlot('Is Borrower Homeowner')
```



50% are borrowers homeowner and 50% are borrowers not homeowner

## Q8: What is the most income range of the borrowers?

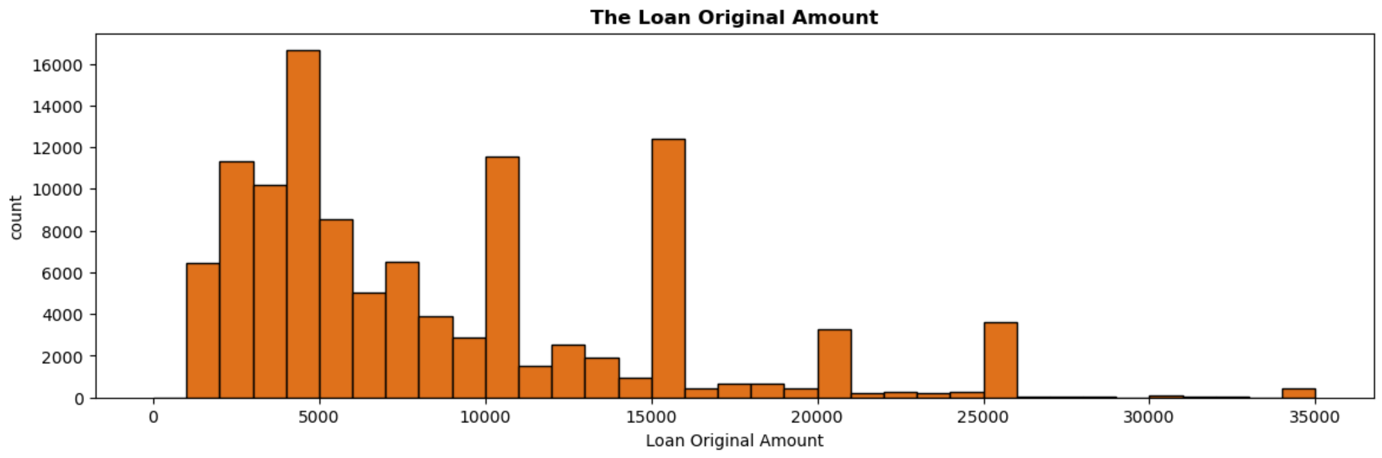
```
In [213…  bar_h('Income Range')
```



The most income range is 25,000-49,999 with 28% then 50,000-74,999 with 27%

## Q9: What is loan original amount?

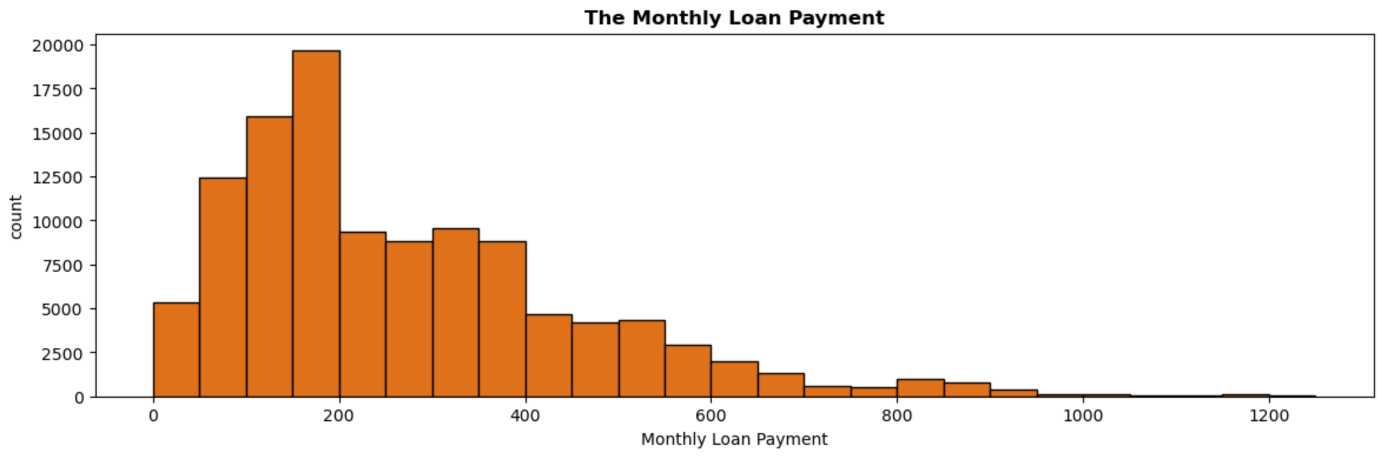
```
In [214…  bins=np.arange(0,loan_n['Loan Original Amount'].max()+1000,1000)
          plt.figure(figsize=[14,4])
          plt.hist(data=loan_n,x='Loan Original Amount',color='#DF711B',bins=bins,edgecolor='black
          plt.xlabel('Loan Original Amount')
          plt.ylabel('count')
          plt.title('The Loan Original Amount',weight='bold');
```



The most loan original amount is 5.000 then 15.000 and 10.000

## Q10: What is the Monthly Loan Payment?

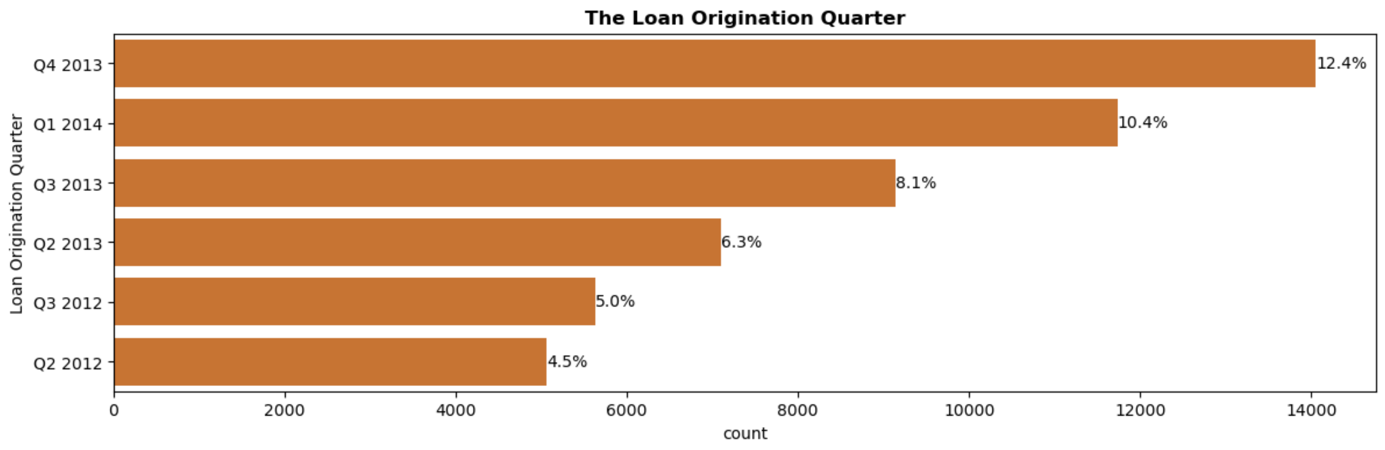
In [215…
```
plt.figure(figsize=[14,4])
bins=np.arange(0,loan_n['Monthly Loan Payment'].max()-1000,50)
plt.hist(data=loan_n,x='Monthly Loan Payment',color='#DF711B',bins=bins,edgecolor='black
plt.xlabel('Monthly Loan Payment')
plt.ylabel('count')
plt.title('The Monthly Loan Payment',weight='bold');
```



The most monthly loan payment between is `$` 100 and `$` 400

## Q11: What is most quarter in which the loan was originated?

In [216…
```
    plt.figure(figsize=[14,4])
    order1=loan_n['Loan Origination Quarter'].value_counts()[0:6].index
    sb.countplot(data=loan_n,y='Loan Origination Quarter',color='#DF711B',order=order1)
    plt.title('The Loan Origination Quarter',weight='bold')

    count_x=loan_n['Loan Origination Quarter'].value_counts()[0:6]
    n_x=loan_n['Loan Origination Quarter'].value_counts().sum()
    for i in range(count_x.shape[0]):
        count=count_x[i]
        per='{:0.1f}%'.format(count*100/n_x)
        plt.text(count+1,i,per,va='center')
```



The most quarter is the fourth quarter of 2013 with 12.4%, then the first quarter of 2014 with 10.4% .

### Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you need to perform any transformations?

> There is the tranformation of this Borrower APR into the form of percentage.

### Of the features you investigated, were there any unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

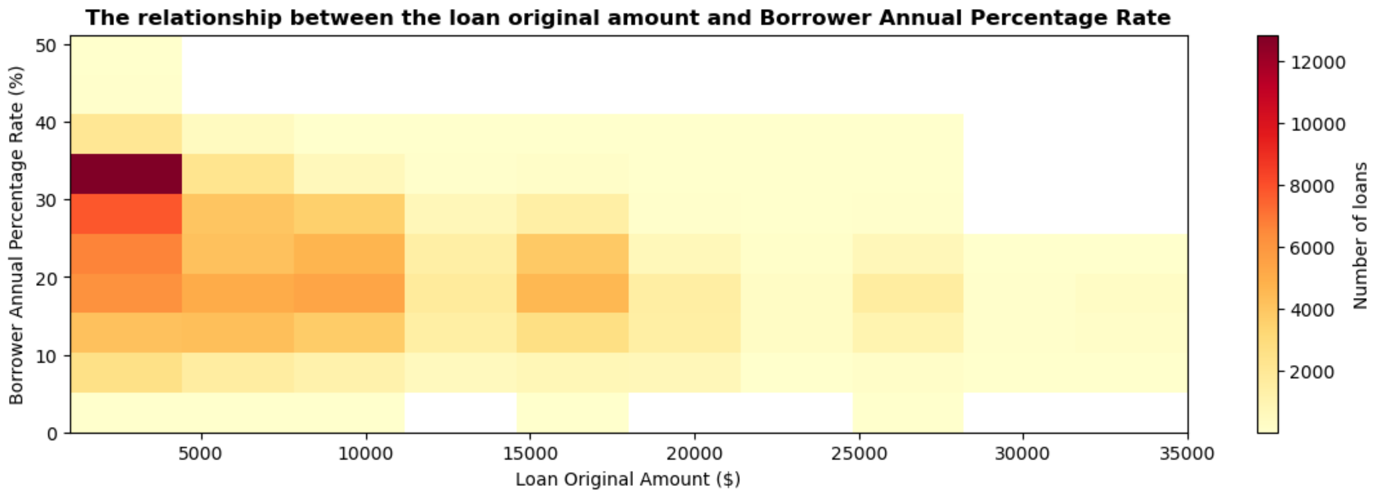
> In Investors I performed log transformation to take a closer look at the data.

## Bivariate Exploration

### Q12: What is relationship between the loan original amount and Borrower Annual Percentage Rate?

In [217…

```
plt.figure(figsize=[14,4])
x_bins=np.arange(1000,loan_n['Loan Original Amount'].max()+2000,2000)
y_bins=np.arange(loan_n['BorrowerAPR_percent'].min(),loan_n['BorrowerAPR_percent'].max()
h2d=plt.hist2d(data = loan_n, x ='Loan Original Amount', y ='BorrowerAPR_percent', cmin=
plt.colorbar(label = 'Number of loans')
plt.title('The relationship between the loan original amount and Borrower Annual Percent
plt.xlabel('Loan Original Amount ($)')
plt.ylabel('Borrower Annual Percentage Rate (%)');
```

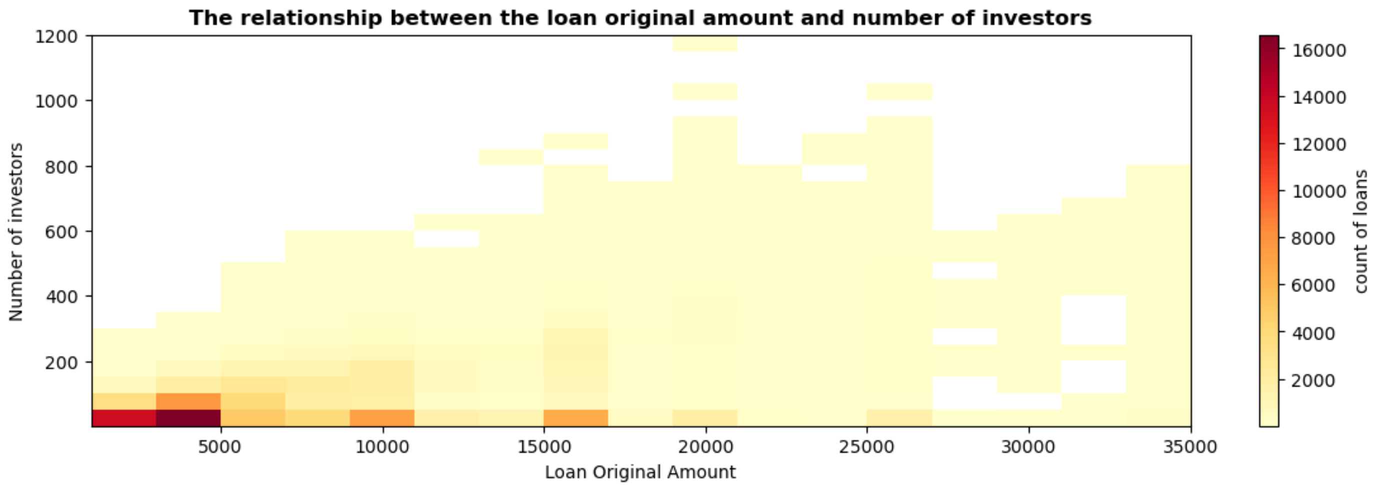


We find that the lower the loan amount, the higher the Borrower Annual Percentage Rate

## Q13:What is the relationship between the loan original amount and number of investors?

In [218…

```
plt.figure(figsize=[14,4])
x_bins=np.arange(1000,loan_n['Loan Original Amount'].max()+2000,2000)
y_bins=np.arange(1,loan_n['Investors'].max()+50,50)
h2d=plt.hist2d(data = loan_n, x ='Loan Original Amount', y ='Investors', cmin=0.5, cmap=
plt.colorbar(label = 'count of loans')
plt.title('The relationship between the loan original amount and number of investors',we
plt.xlabel('Loan Original Amount')
plt.ylabel('Number of investors');
```

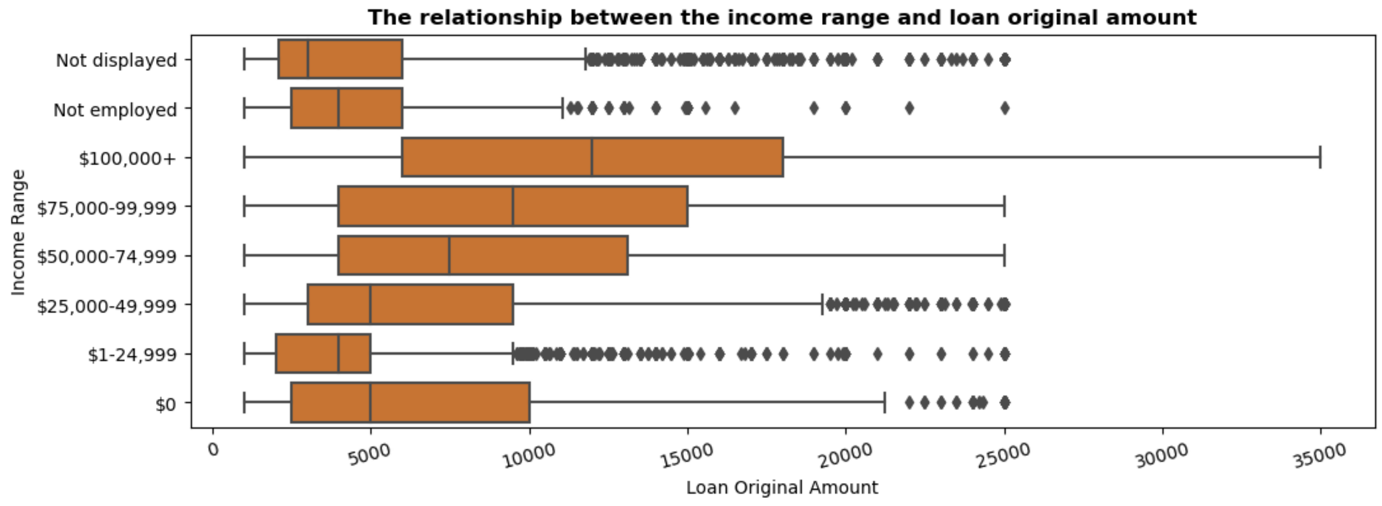


The most loan original amount less than of 5000 `$` with less than 100 investors.

## Q14: What is the relationship between the income range and loan original amount?

In [219…

```
plt.figure(figsize=[12,4])
order_b=['Not displayed','Not employed','$100,000+','$75,000-99,999','$50,000-74,999','$
sb.boxplot(data = loan_n, x = 'Loan Original Amount', y ='Income Range' ,color = '#DF711
plt.title('The relationship between the income range and loan original amount', weight='
plt.xticks(rotation=15);
```



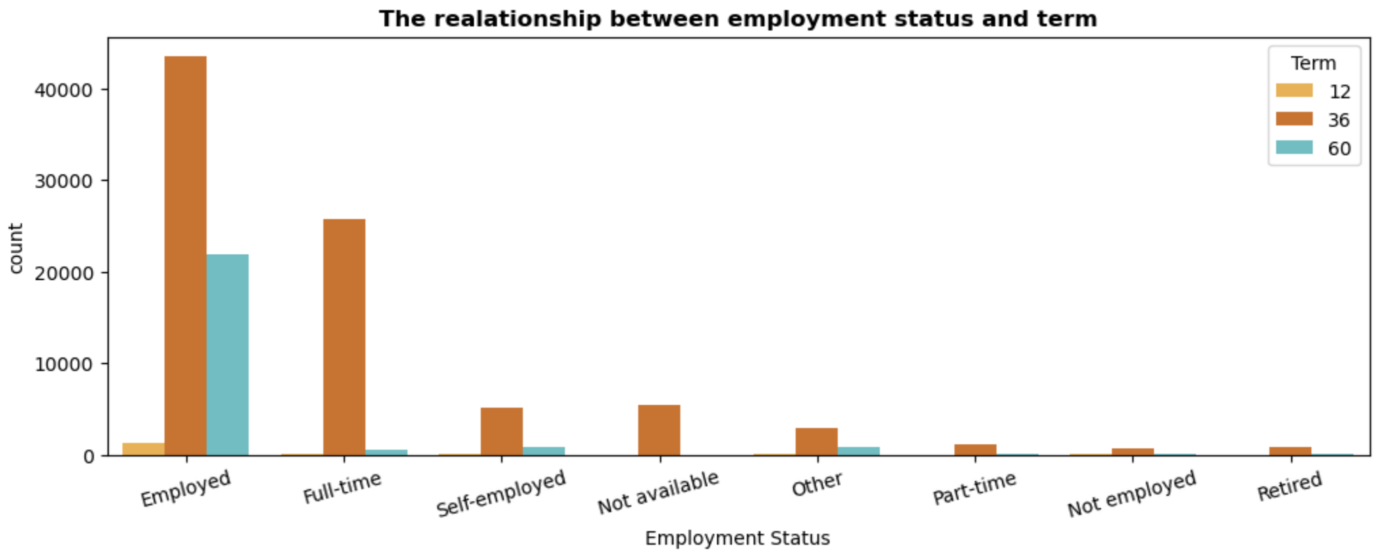
We find that the higher the range income, the higher the loan amount.

People with low incomes and those who have no income cannot borrow loans in high amounts.

## Q15: What is the realationship between employment status and term

In [220…

```
plt.figure(figsize=[12,4])
order_c=loan_n['Employment Status'].value_counts().index
sb.countplot(data = loan_n, x = 'Employment Status', hue = 'Term',
             palette=['#FFB740','#DF711B','#64C9CF'],order=order_c)
plt.title('The realationship between employment status and term',weight='bold')
plt.xticks(rotation=15);
```

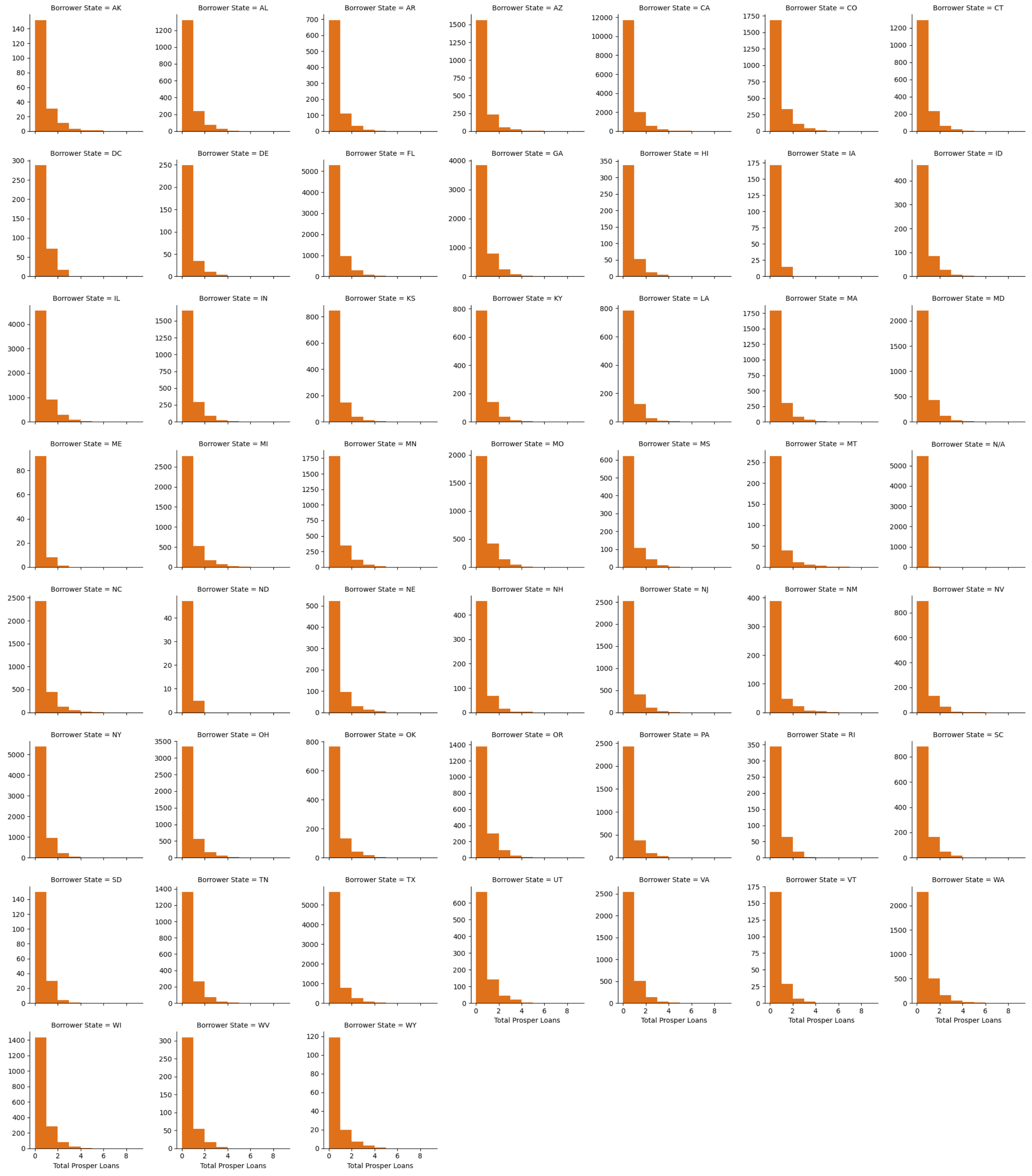


We find that the 36 term is the most frequent, and we find that most of the borrowers are employed in the 36 term.

## Q16: What is the number of total prosper loans for each borrower state?

In [221…

```
bins_edge=np.arange(0, loan_n['Total Prosper Loans'].max()+2,1)
g = sb.FacetGrid(data = loan_n, col = 'Borrower State', col_wrap=7, sharey=False)
g.map(plt.hist, "Total Prosper Loans",bins=bins_edge,color='#DF711B');
```

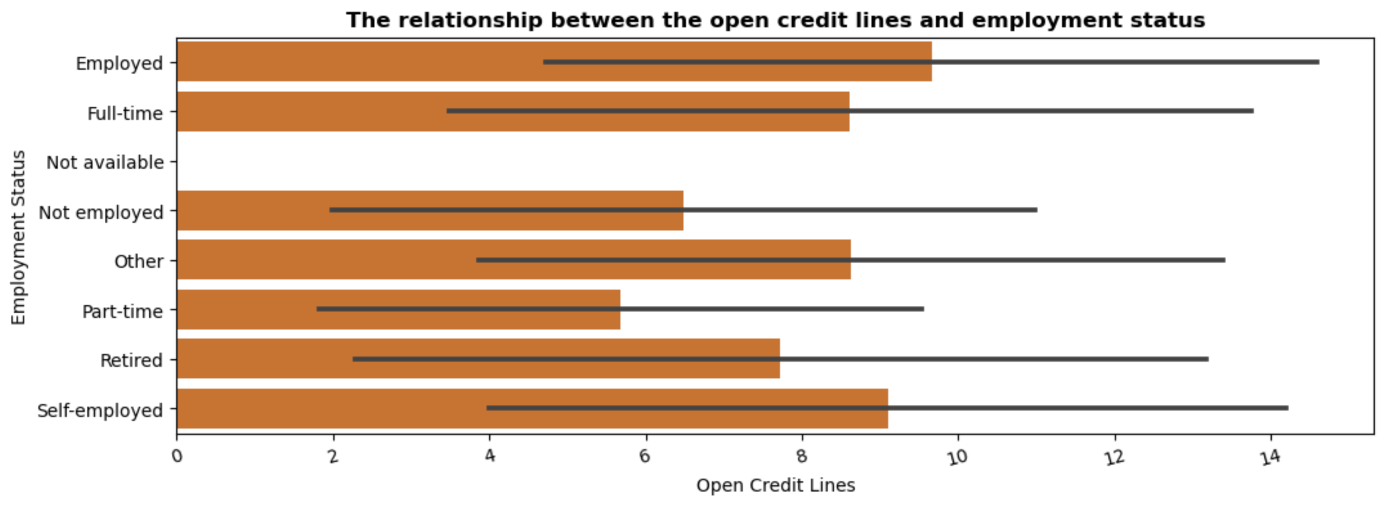


### Most states don't have any loans before this time

## Q17: What is the relationship between open credit lines and employment status?

In [222…

```
plt.figure(figsize=[12,4])
sb.barplot(data = loan_n, x = 'Open Credit Lines', y ='Employment Status' ,color = '#DF7
plt.title('The relationship between the open credit lines and employment status', weight
plt.xticks(rotation=15);
```



We find the height values in open credit lines are for employed and self-employed

## Talk about some of the relationships you observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

> In the relationship between the loan original amount and Borrower Annual Percentage Rate We find that the lower the loan amount, the higher the borrower APR.
>
> In the relationship between the income range and loan original amount We find that the higher the range income, the higher the loan amount. People with low incomes and those who have no income cannot borrow loans in high amounts.
>
> In the relationship between employment status and term We find that the 36 term is the most frequent, and we find that most of the borrowers are employed in the 36 term.

## Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

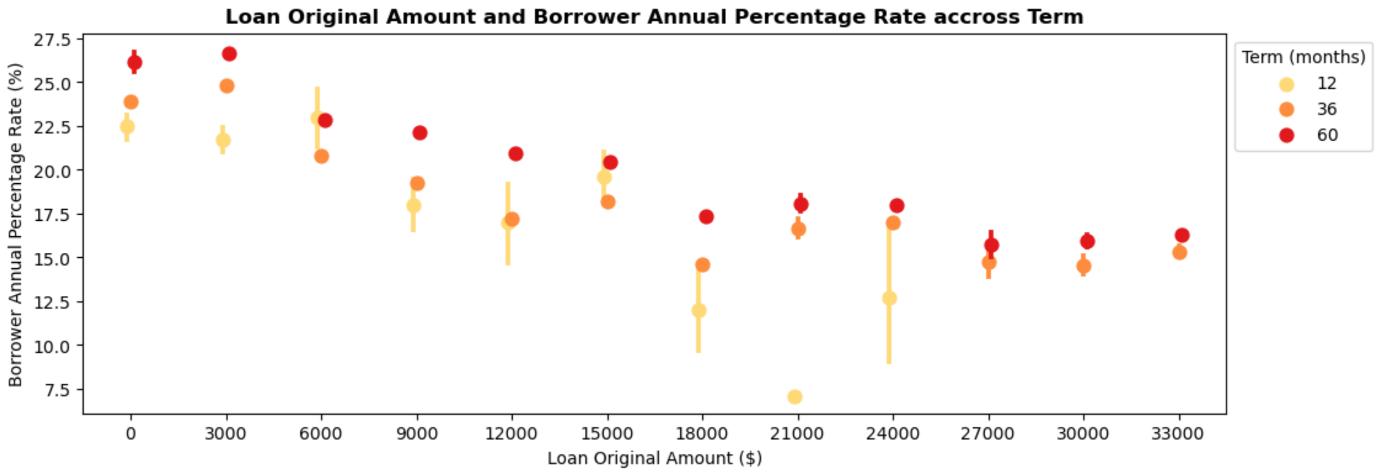
> In the relationship between total prosper loans for each borrower state we find most states don't have any loans before this time
>
> In the relationship between open credit lines and employment status We find the height values in open credit lines are for employed and self-employed

## Multivariate Exploration

### Q18:What is Loan Original Amount and Borrower Annual Percentage Rate across Term?

In [223…

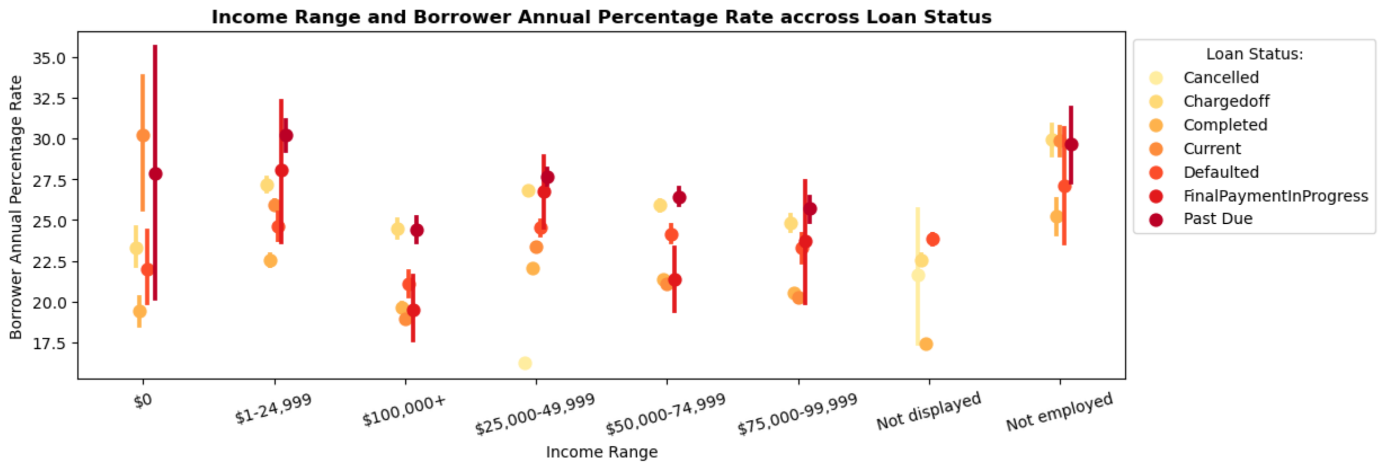
```
loan_n['Loan Amount']=((loan_n['Loan Original Amount']//3000)*3000)
loan_n['Loan Amount'].value_counts().sort_index()

plt.figure(figsize = [12, 4])
ax=sb.pointplot(data=loan_n,x='Loan Amount',y = 'BorrowerAPR_percent' ,hue='Term' ,palet
                linestyles="",dodge=True)
ax.set_yticklabels([],minor=True)
plt.legend(title='Term (months)',bbox_to_anchor=(1,1),loc="upper left")
plt.title('Loan Original Amount and Borrower Annual Percentage Rate accross Term',weight
plt.ylabel('Borrower Annual Percentage Rate (%)')
plt.xlabel('Loan Original Amount ($)');
```



We find that the highest Borrower Annual Percentage Rate is for the less the Loan Original Amount with 60 months.

## Q19: What is the Income Range and Borrower Annual Percentage Rate across Loan Status?

In [224…
```
plt.figure(figsize = [12, 4])
ax=sb.pointplot(data=loan_n,x='Income Range',y = 'BorrowerAPR_percent' ,hue='Loan Status
                palette='YlOrRd',linestyles="",dodge=True)
ax.set_yticklabels([],minor=True)
plt.legend(title='Loan Status:',bbox_to_anchor=(1,1),loc="upper left")
plt.title('Income Range and Borrower Annual Percentage Rate accross Loan Status',weight=
plt.ylabel('Borrower Annual Percentage Rate')
plt.xlabel('Income Range')
plt.xticks(rotation=15);
```

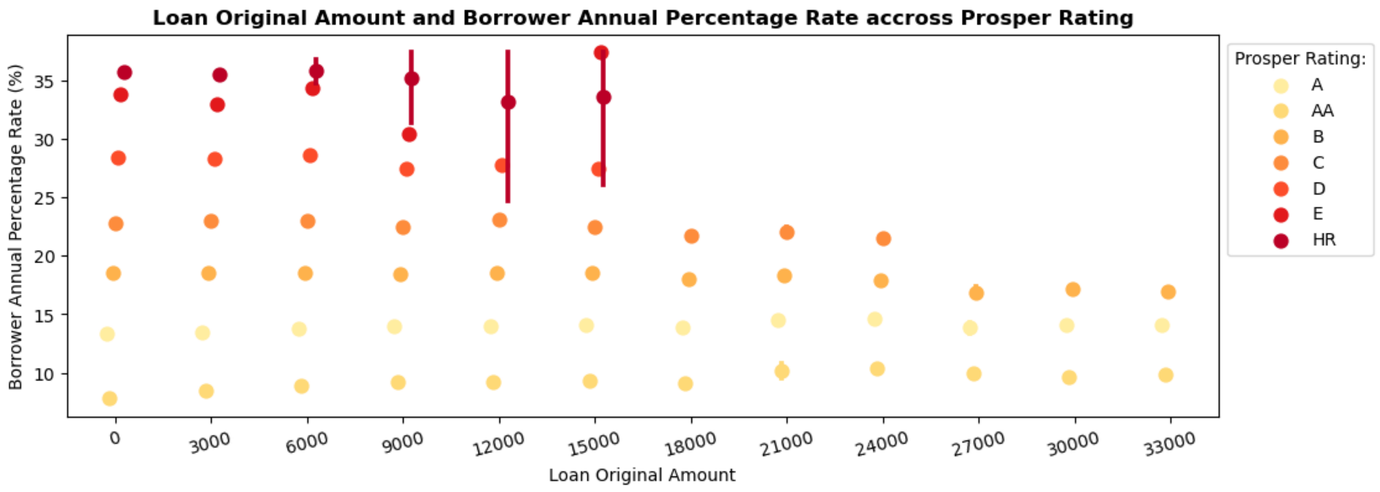


We find that the highest Borrower Annual Percentage Rate for each income range is `past due`

## Q20: What is Loan Original Amount and Borrower Annual Percentage Rate across Loan Status?

In [225…
```
plt.figure(figsize = [12, 4])
ax=sb.pointplot(data=loan_n,x='Loan Amount',y = 'BorrowerAPR_percent' ,hue='Prosper Rati
                palette='YlOrRd',linestyles="",dodge=True)
```

```
ax.set_yticklabels([],minor=True)
plt.legend(title='Prosper Rating:',bbox_to_anchor=(1,1),loc="upper left")
plt.title('Loan Original Amount and Borrower Annual Percentage Rate accross Prosper Rati
plt.ylabel('Borrower Annual Percentage Rate (%)')
plt.xlabel('Loan Original Amount')
plt.xticks(rotation=15);
```

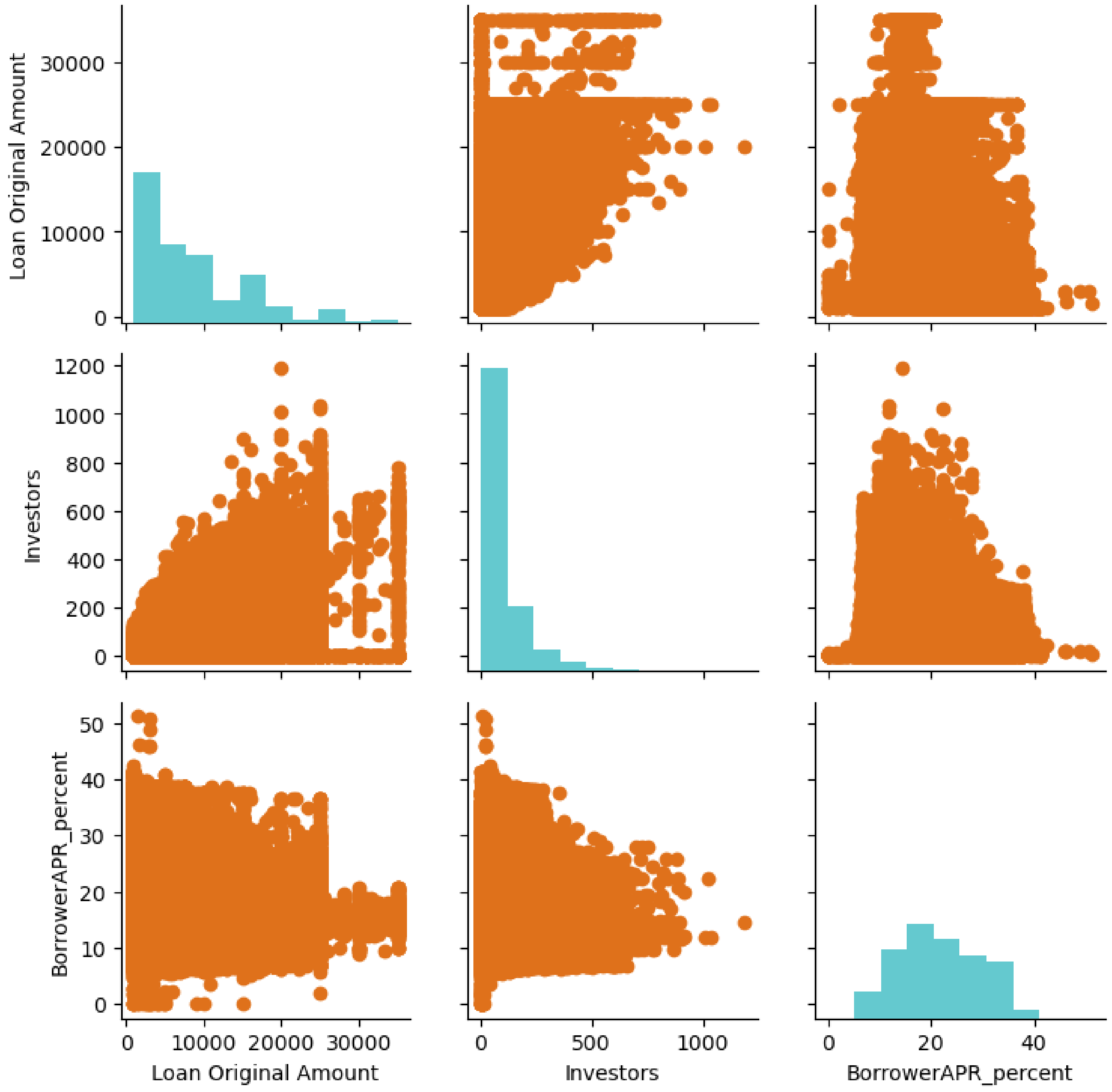


**We find highest risk rating in high Borrower Annual Percentage Rate**

## Q21: What is the relationship between Loan Original Amount and Investors and Borrower Annual Percentage Rate?

In [226…

```
g = sb.PairGrid(data = loan_n, vars = ['Loan Original Amount', 'Investors',
                                        'BorrowerAPR_percent'],palette='Oranges')
g.map_diag(plt.hist,color='#64C9CF')
g.map_offdiag(plt.scatter,color='#DF711B');
```



### Talk about some of the relationships you observed in this part of the investigation. Were there features that strengthened each other in terms of looking at your feature(s) of interest?

> In the relationship between Loan Original Amount and Borrower Annual Percentage Rate across Term We find that the highest Borrower Annual Percentage Rate is for the less the Loan Original Amount.
>
> In the relationship Loan Original Amount and Borrower Annual Percentage Rate across Loan Status

### Were there any interesting or surprising interactions between features?

> In the relationship between the Income Range and Borrower Annual Percentage Rate across Loan Status We find that the highest Borrower Annual Percentage Rate for each income range is `past due`

## Conclusions

> • The most percentage in Borrower Annual Percentage Rate is 20%.
>
> • The most length of the loan is 36 months with 77%.
>
> • The most status of the loan is Current with 49%.
>
> • The most employment status of the borrowers is employed with 60% then full-time with 23%.
>
> • 50% of category of the listing that the borrower selected is Debt Consolidation.
>
> • Most investors under 100.
>
> • 50% are borrowers homeowner and 50% are borrowers not homeowner.
>
> • The most income range is 25,000-49,999 with 28% then 50,000-74,999 with 27%.
>
> • The most loan original amount is 5.000 then 15.000 and 10.000.
>
> • The most monthly loan payment between is `$` 100 and `$`400.
>
> • The most quarter is the fourth quarter of 2013 with 12.4%, then the first quarter of 2014 with 10.4%.
>
> • The lower the loan amount is the higher the Borrower Annual Percentage Rate.
>
> • The most loan original amount less than of 5000 `$` with less than 100 investors.
>
> • The higher the range income is the higher the loan amount and people with low incomes and those who have no income cannot borrow loans in high amounts.
>
> • The 36 term is the most frequent, and we find that most of the borrowers are employed in the 36 term.
>
> • Most states don't have any loans before this time.
>
> • The height values in open credit lines are for employed and self-employed.
>
> • The highest Borrower Annual Percentage Rate is for the less the Loan Original Amount with 60 months.
>
> • The highest risk rating in high Borrower Annual Percentage Rate.
>
> • The highest Borrower Annual Percentage Rate for each income range is past due

## Resources:

```
https://www.adamsmith.haus/python/answers/how-to-print-an-entire-pandas-
dataframe-in-python
https://matplotlib.org/stable/tutorials/colors/colormaps.html#classes-of-
colormaps
https://www.geeksforgeeks.org/matplotlib-pyplot-hist2d-in-python/
```

https://stackoverflow.com/questions/57417970/how-to-set-custom-colors-on-a-count-plot-in-seaborn
https://seaborn.pydata.org/generated/seaborn.PairGrid.html
https://seaborn.pydata.org/tutorial/color_palettes.html
https://www.doughroller.net/resources/reviews/prosper-review/